ECU Fingerprinting through Parametric Signal Modeling and Artificial Neural Networks for In-vehicle Security against Spoofing Attacks

Azeem Hafeez
CECS Department
University of Michigan
Dearborn, USA
azeemh@umich.edu

Kenneth Topolovec
CECS Department
University of Michigan
Dearborn, USA
ktopolov@umich.edu

Selim Awad
CECS Department
University of Michigan
Dearborn, USA
sawad@umich.edu

***Abstract*—Fully connected autonomous vehicles are more vulnerable than ever to hacking and data theft. The controller area network (CAN) protocol is used for communication between in-vehicle control networks (IVN). The absence of basic security features of this protocol, like message authentication, makes it quite vulnerable to a wide range of attacks including spoofing attacks. As traditional cybersecurity methods impose limitations in ensuring confidentiality and integrity of transmitted messages via CAN, a new technique has emerged among others to approve its reliability in fully authenticating the CAN messages. At the physical layer of the communication system, the method of fingerprinting the messages is implemented to link the received signal to the transmitting electronic control unit (ECU). This paper introduces a new method to implement the security of modern electric vehicles. The lumped element model is used to characterize the channel-specific step response. ECU and channel imperfections lead to a unique transfer function for each transmitter. Due to the unique transfer function, the step response for each transmitter is unique. In this paper, we use control system parameters as a feature-set, afterward, a neural network is used transmitting node identification for message authentication. A dataset collected from a CAN network with eight-channel lengths and eight ECUs to evaluate the performance of the suggested method. Detection results show that the proposed method achieves an accuracy of 97.4% of transmitter detection.**

***Index Terms*—Intrusion detection system (IDS), electronic control unit (ECU), controller area network (CAN), machine learning (ML), artificial neural network (ANN), performance matrix (PM), receiver operating characteristic (ROC), Lumped element model (LEM).**

I. Introduction

In the modern electric vehicle, CAN is used as a reliable, robust, and simple network protocol for in-vehicle communication. CAN is widely applicable in automotive industry applications, and it makes the transmission of the messages between the ECUs and the devices more organized and controllable. This protocol lacks message authentication, because the sender information in the packet is missing, which makes it vulnerable to spoofing attacks. In the past few years, security and safety are some of the major concerns for the vehicle industry. In 2015, Miller et. al. [1] killed the vehicle engine in the middle of the highway, and they were able to take control of the vital functionalities of the vehicle like engine, braking unit, etc. The purpose of this experiment [1] was to convey a message to the stakeholders that the implementation of security in the electric car is a vital issue. The cybersecurity industry as a whole is projected to grow substantially in the coming years, as depicted by *Fig.* 1. Companies continue to increase investment in cybersecurity to keep up with the increase in worldwide technology and connectivity.

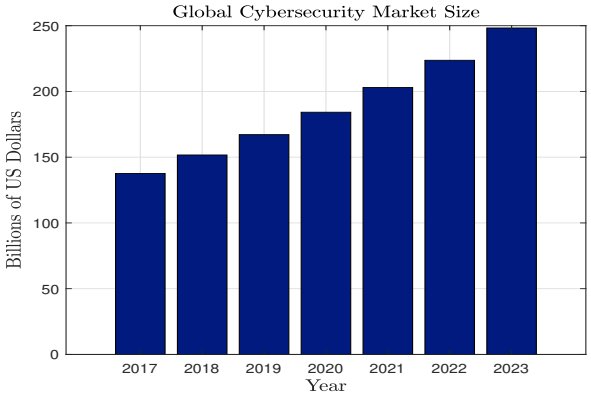

Fig. 1: *Projected Growth of Global Cybersecurity* [2]

With the advent of autonomous vehicles (AVs), interest in the safety and security of vehicles is increasing. These autonomous vehicles are considered a breakthrough in modern technology; however, they come with cyber vulnerability risks. These risks must be addressed before autonomous vehicles become readily available in the market. As shown in *Fig.* 2, this could be right around the corner. Autonomous vehicles could already comprise 15% of vehicle sales by the end of the next decade, provided that the growth of autonomy encounters little legal and infrastructural resistance [3].

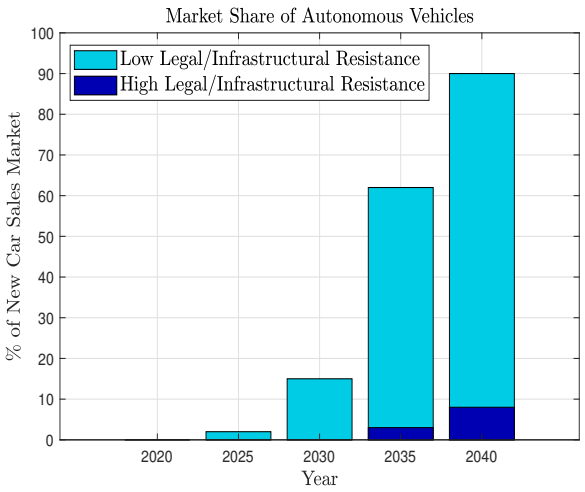

Fig. 2: *Projected Market Share of Autonomous Vehicles in New Car Sales* [3]

Existing state-of-the-art countermeasures against spoofing and impersonation attacks on in-vehicle CAN networks can be divided into (i) message authentication code (MAC) based approaches [4]–[14], and (ii) intrusion detection based approaches [15]–[40]. Intrusion detection based approaches can be further subdivided into, (a) fingerprinting-based approach [15]–[17], (b) parameter monitoring-based approach [20]–[24], (c) information theory-based approach [25]–[27], and (d) machine learning-based approach [28]–[34]. Current solutions [14], [41] for CAN communication are limited in their ability and scope as they are unable to identify the transmitting ECU responsible for the received packet. This paper presents a novel approach to identify the source ECU of a CAN signal to authenticate the message. The idea of fingerprinting the source is also used for other electronic devices such as microphones [42], [43]. In CAN communication, when a transmitter sends a message, the expected signal at the receiver is a rectangular waveform. However, due to channel imperfections, it is not a perfect rectangular waveform. The channel has a transfer function, and the input to this channel is a rectangular wave, which can be modeled as a step function. Hence, the input to the receiver can be quantified as the step response of the channel, through which the signal propagates. *Fig.* 3 shows the step response of the channel to the CAN signal. It can be observed that the received signal is not the same as the ideal signal. This paper proposes a new technique to address the limitations of the existing state of the art CAN security methods by exploiting the channel based step response; and then finding the transient, and steady-state parameters like overshoot, peak time, settling time, peak value, steady-state value, damping ratio and natural frequency. Furthermore, these parameters are used as inputs to a neural network classifier using supervised learning to authenticate a message by identifying its transmitter.

Contributions: The main contributions of this paper are:

- Modeling of channel-specific step response for CAN signals
- Channel-specific step response uniqueness analysis
- Propose a reliable framework for identifying the source ECU of any given CAN message

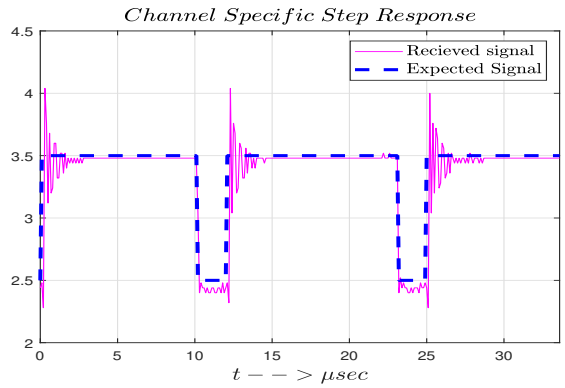

Fig. 3: *Channel Specific Step response*

The rest of this paper is structured as follows: Section II is the discussion of the CAN communication protocol; section III provides mathematical modeling of channel-specific distortion and its uniqueness analysis; it also provides mathematical modeling of the system; section IV outlines the experimental setup, data collection, results, and analysis; and section V provides a conclusion and future direction.

II. CAN Protocol Overview

CAN protocol was invented by Robert Bosch in 1986. It is a robust, convenient and reliable in-vehicle communication system that consists of nodes that share the same bus and are wired to send messages in the form of bits, 0's and 1's [44]. The arbitration scheme decides that the node with the lower ID is more dominant and has a higher priority in transmitting data than other nodes, which makes CAN bus perfectly usable as a real-time communication bus [44]. CAN protocol is considered one of the strongest and most dependable protocols in automotive communication, but it lacks some essential security features; namely, the absence of

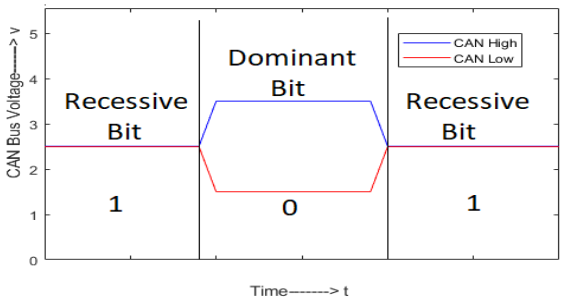

Fig. 4: *CAN Signalling*

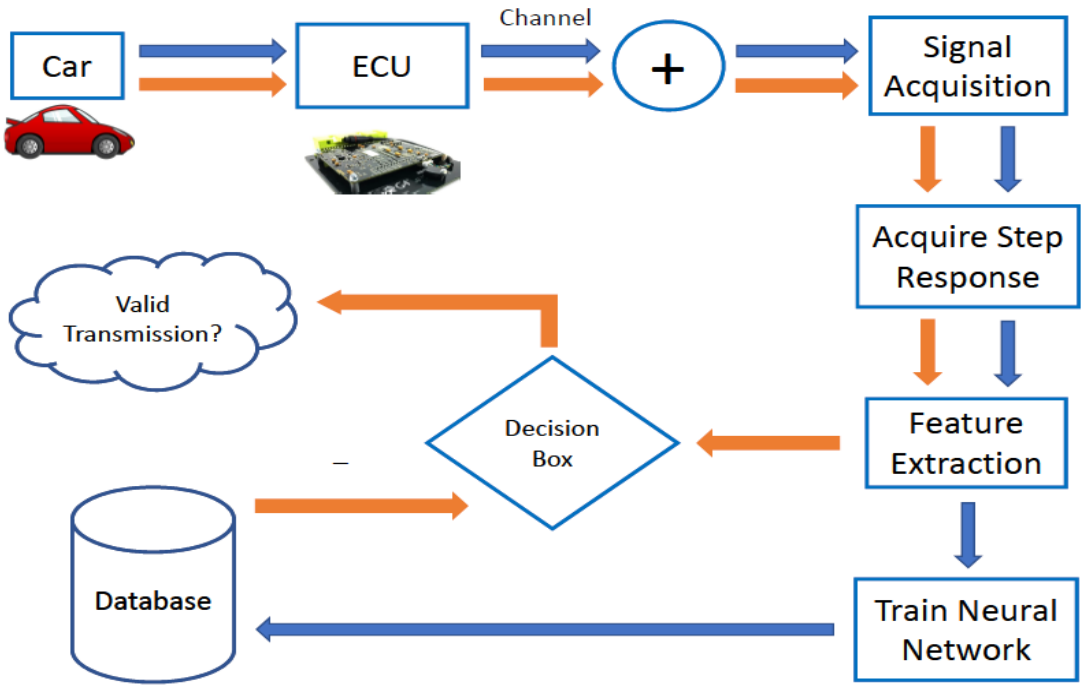

Fig. 5: Block diagram of transmitter identification system

the sender identity and the message authentication [4]–[14]. In CAN communication, a logical 0 is considered the dominant bit and a logical 1 is labeled the recessive bit. $Fig.$ 4 shows the voltage which appears on CAN-High ($CANH$) and CAN-Low ($CANL$) for the dominant as well as recessive bit. In order to transmit a dominant bit, a voltage of $3.5V$ is transmitted on $CANH$ for a bit interval; when a recessive bit is sent, a voltage of $2.5V$ is sent from $CANH$ for a bit interval. Similarly, for a dominant bit, a voltage of $1.5V$ is transmitted on $CANL$ for a bit interval; when a recessive bit is sent, a voltage of $2.5V$ is sent from $CANL$ for a bit interval.

III. System and Mathematical Modelling

$Fig.$ 5 shows the architecture of the method presented for transmitter identification for message authentication. This architecture is deployed at fingerprinting ECU $E_{(FP)}$ as shown in $Fig.$ 6. In this architecture, the subnet has a total of N ECUs connected to N channels, where any ECU can be represented as $E_{(i)}$, $\lambda = \{1, 2, \ldots, N\}$, where $i \in \lambda$. The fingerprinting ECU $E_{(FP)}$ sniffs the signal transmitted by all transmitters, and the response due to each channel at the $E_{(FP)}$ is then used for feature extraction to generate the feature vector $\mathbf{X_{(r)}} = \{x^{(1)}, x^{(2)}, \ldots, x^{(M)}\}$ for $E_{(i)}$, where M represents the number of features, $r \in R$ and R is the total number of records. The feature vector is then passed to a neural network architecture, which is pre-trained on R records for message authentication.

A. Channel Modeling:

The signal transmitted by $E_{(i)}$ represented by $x_{(i)}(t)$ propagates through the i^{th} channel ($h_{(i)}(t)$), the channel behaves like a linear time-invariant system, the output of the channel is represented as $y^{(a)}_{(i)}(t)$. The relationship between $x_{(i)}(t)$ and $y^{(a)}_{(i)}(t)$ can be represented as eq. (1). Here, '$*$' operator represents convolution.

$$y^{(a)}_{(i)}(t) = x_{(i)}(t) * h_{(i)}(t). \tag{1}$$

Shown in the right column of $Fig.$ 7 is the equivalent circuit of an infinitesimally small piece of a transmission line. According to the lumped element model (LEM), the transmission line is represented as a series resistance (R'), series Inductance (L'), a parallel Conductance (G') and a parallel capacitance (C'). Shown in the left column of $Fig.$ 7 is the physical structure of the CAN channel. In this structure, D is the distance between 2 wires, d is the diameter of the wires. Let μ be the permeability, and σ be the conductivity of the copper. The ideal line parameters R',

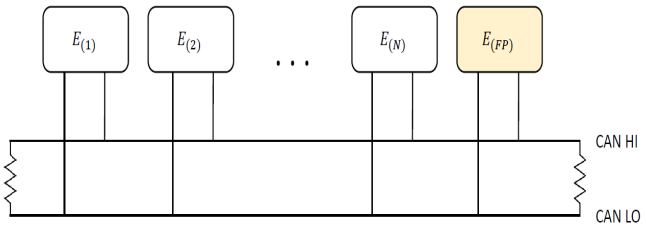

Fig. 6: $Message\ authentication\ by\ E_{(FP)}$

C', L' and G' [45] can be expressed in the following equations:

$$R' = \frac{2R_s}{\pi d}, \qquad L' = \frac{\mu}{\pi} \ln \left\{ \frac{D}{d} + \sqrt{(\frac{D}{d})^2 - 1} \right\}$$
$$C' = \frac{\pi \epsilon}{\ln \left\{ \frac{D}{d} + \sqrt{(\frac{D}{d})^2 - 1} \right\}}, \quad G' = \frac{\pi \sigma}{\ln \left\{ \frac{D}{d} + \sqrt{(\frac{D}{d})^2 - 1} \right\}}$$

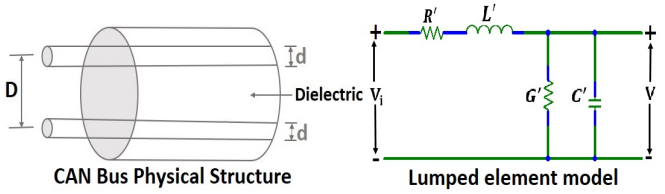

Fig. 7: *LEM for transmission line*

We hypothesize that due to manufacturing imperfections, the length of the cable, $h_{(i)}(t)$ is unique for every channel, so the step response of each channel is unique. In this research, we have used eight channels. Given the circuit is shown in *Fig.* 7, we can find the transfer function relating V_i to V in the Laplace domain. Let $H_i(s)$ represent the Laplace transform of $h_{(i)}(t)$, which is represented as eq. (2).

$$H_i(s) = \frac{1}{L'C'} \left[\frac{1}{s^2 + \frac{R'C'+L'G'}{L'C'} s + \frac{1+G'R'}{L'C'}} \right] \tag{2}$$

B. Signal Acquisition:

The $E_{(FP)}$ acquires the analog signal $y_{(i)}^{(a)}(t)$ generated by $E_{(i)}$ and transmitted through $h_{(i)}(t)$ and converts this signal into digital signal $y_{(i)}(n)$ as represented in eq. (3).

$$y_{(i)}(n) = y_{(i)}^{(a)}(t)|_{t=nT_s}, T_s = 0.5 \times 10^{-9} \tag{3}$$

Where T_s represents sampling time of $0.5\,nsec$ and the sampling rate of $2\,GSa/sec$ for the signal. The reason to generate $y_{(i)}(n)$ is that the $y_{(i)}^{(a)}(t)$ occurs at infinite instants of time, thus demands large memory to get stored. However, the $E_{(FP)}$ has limited memory, therefore, the analog-to-digital conversion is performed.

C. Step Response Acquisition:

After the signal acquisition, the next step is to acquire a step response of the system. The motivation behind finding the step response is to find control system parameters that are used as a feature set.

1) Pulse Isolation*:* After digitization, an algorithm is used to isolate individual dominant bits from the $CANH$ bus. This section will focus purely on the detection of the rising and falling edges of each $CANH$ pulse, such that these pulses can be extracted. Individual pulses are needed so we can look at an individual step response and model only that single step response in terms of its control parameters. For the

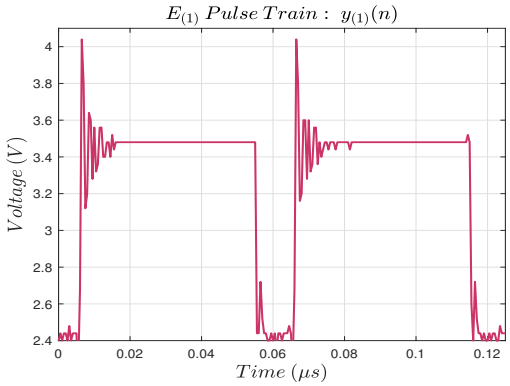

Fig. 8: *ECU's Digitally Sampled Pulse Train*

following, assume that a single ECU's signal which passes through channel i is being processed, $y_{(i)}(n)$, which contains X samples in total and K pulses. A small subset of samples from the digital pulse train $y_{(i)}(n)$ is shown in *Fig.* 8.

To isolate the individual pulses shown in *Fig.* 8 to extract their control parameters, we first must design an algorithm to locate the edges of each dominant bit. For this edge detection, first, we filter $y_{(i)}(n)$ using a P-sample digital moving average filter to smooth oscillations, reducing the likelihood of false edge detection. By suppressing high-amplitude oscillations with this filter, there is less chance to detect oscillations as edges. For our purposes, we used $P = 10$. The new moving average signal is shown in *Fig.* 9 and is mathematically represented as a difference equation in eq. (4). Allow $y_{(i)}^{(MA)}(n)$ to represent the moving average filtered output of $y_{(i)}(n)$.

$$y_{(i)}^{(MA)}(n) = \frac{1}{P} \sum_{i=1}^{P} y_{(i)}(n-i) \tag{4}$$

Next, the signal is converted to an ideal digital signal by using a threshold of $3V$ to classify individual samples as being part of either a dominant or a recessive bit as shown in eq. (5).

$$y_{(i)}^{(D)}(n) = \begin{cases} 0, & y_{(i)}^{(MA)}(n) < 3 \\ 1, & 3 \leq y_{(i)}^{(MA)}(n) \end{cases} \tag{5}$$

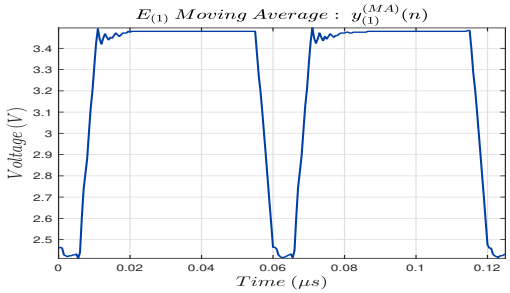

Fig. 9: *Moving Average Pulse Train*

Fig. 10 shows this digitally-thresholded signal. Allow $y_{(i)}^{(D)}(n)$ to be the digitally-thresholded signal of $y_{(i)}^{(MA)}(n)$.

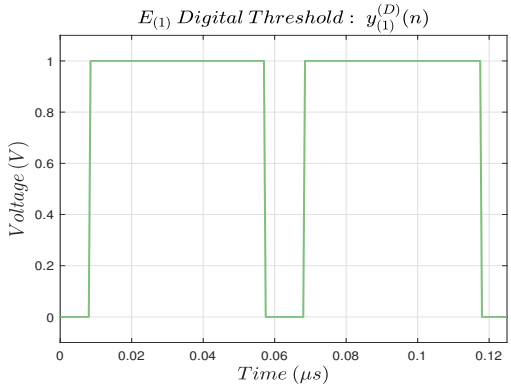

Fig. 10: *Digitally Thresholded Pulse Train*

The ideal digital signal contains no oscillations, so we can now find the digital derivative of the digital signal $y_{(i)}^{(D)}(n)$ using the first difference, as shown in eq. (6), to find the rising and falling edges.

$$\frac{dy}{dn}y_{(i)}^{(D)}(n) = y_{(i)}^{(D)}(n) - y_{(i)}^{(D)}(n-1), \quad n = 0, 1, ..., X-1 \tag{6}$$

All rising edges are now shown by a positive impulse with some delay, in samples, while all falling edges are shown by a negative impulse, as shown in *Fig.* 11. All non-edges assume a value of zero. We extract indices for rising edges and falling edges as shown in eq. (7). Allow n_{rise} and n_{fall} to represent the sample indices of the detected rising and falling edges respectively.

$$n_{rise} = n|\frac{dy}{dn}y_{(i)}^{(D)}(n) > 0 \quad n_{fall} = n|\frac{dy}{dn}y_{(i)}^{(D)}(n) < 0 \tag{7}$$

2) Outlier Removal*:* In CAN, it is possible for an ECU to occasionally transmit erroneous pulses which do not exhibit similar characteristics to a typical dominant bit from this ECU. It is necessary to remove these anomalies because their

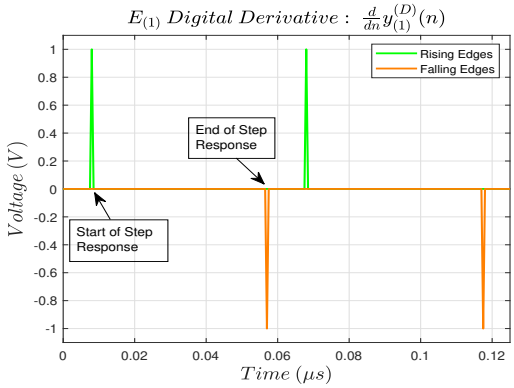

Fig. 11: *Rising and Falling Edges of Pulse Train*

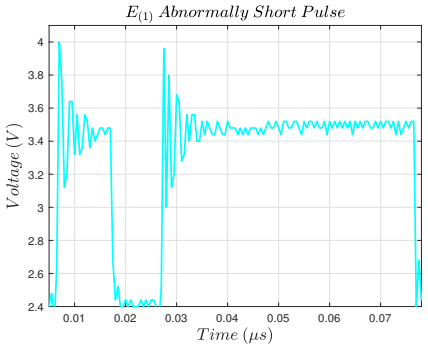

Fig. 12: *Abnormally Short Pulse*

signatures will be vastly different from the typical signature of their respective ECU and channel combinations. One possible anomaly scenario is when a single $CANH$ dominant bit is significantly shorter than the typical dominant bit, which may produce anomalous steady-state parameter calculations, since the pulse may not have sufficient time to settle and reach its steady-state. Another anomaly is when a single dominant bit is sent, followed by a second dominant bit that comes too early. This often results in an abnormally long pulse and strange transient behaviors. These short and long pulse behaviors are demonstrated in *Fig.* 12 and *Fig.* 13 respectively.

Pulses similar to those shown in *Fig.* 12 and *Fig.* 13 can be treated as outliers, and will not be considered at all for training or testing the classification model, or for deployment. In effect, we block the fingerprinting ECU from making any decisions about the authenticity of these outlier pulses. To remove these outliers, we formulate the following algorithm. Allow $\mathbb{E}(T)$ to represent the expected pulse length of a typical CAN pulse, in seconds. Also, allow Δt to represent some tolerance, in seconds. We choose to only keep pulses of length T such that the condition is shown in eq. (8) is obeyed.

$$\mathbb{E}(T) - \Delta t < T < \mathbb{E}(T) + \Delta t \tag{8}$$

Since we currently have indices n_{rise} and n_{fall} from eq. (7), where $n_{rise}^{(k)}$ and $n_{fall}^{(k)}$ correspond to the rising and falling edge sample indices of pulse k, we check through all pulses, compute their length $T^{(k)}$ in seconds, and then decide to keep

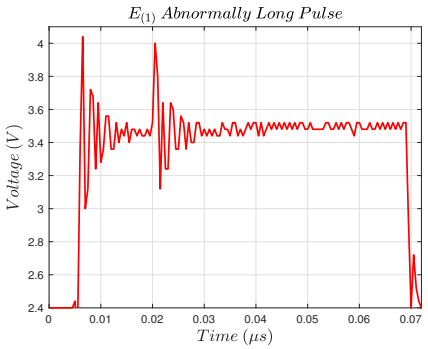

Fig. 13: *Abnormally Long Pulse*

or discard the pulse as an anomaly using the rule discussed in eq. (8):

for $1 \leq k \leq K$ **do**
  $T^{(k)} = n_{fall}^{(k)} - n_{rise}^{(k)}$
  if $\mathbb{E}(T) - \Delta t < T^{(k)} < \mathbb{E}(T) + \Delta t$ **then**
    Keep $n_{fall}^{(k)}$ and $n_{rise}^{(k)}$
  else
    Discard $n_{fall}^{(k)}$ and $n_{rise}^{(k)}$
  end if
end for

After removing indices corresponding to the outliers, we can extract the k^{th} individual pulse of $E_{(i)}$, denoted by $y_{(i,k)}(n)$, using the indices n_{rise} and n_{fall} which have not been discarded. An example of a typical single pulse is shown in $Fig.$ 14.

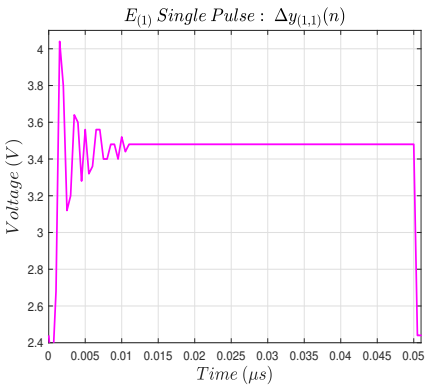

Fig. 14: $Single\ Pulse$

D. Feature Extraction:

After outlier removal, we perform feature extraction to represent each pulse as a collection of control-theory-based parameters. This reduces a relatively high-dimensional input into a very low-dimensional representation, which allows for a lower-complexity model. This low model of complexity is computationally desirable. The parameters used for this study include peak time, T_p, percent overshoot, $\%OS$, settling time, T_s, and steady-state value, V_{ss}. Peak time and percent overshoot describe a second-order system's transient response while settling time and steady-state value describe the system's steady-state behavior. These parameters are described below.

Peak Time: The time required for a step response to reach its peak value from the time that it first begins to rise.

Percent Overshoot: The percentage by which the step response's peak value exceeds the response's value at steady-state.

Settling Time: The amount of time it takes to form the time that the step response peaks until it has settled within some tolerance of its steady-state value.

Steady-State Value: The value that a step response assumes after all transient oscillation has diminished.

Note that the second-order system is described in the Laplace domain by the standard equation shown in eq. (9).

$$H(s) = \frac{K\omega_n^2}{s^2 + 2\zeta\omega_n + \omega_n^2} \tag{9}$$

The three main parameters in this standard form are known as the natural frequency (ω_n), the damping ratio (ζ), and the D.C. gain of the system, K. In this application, K will not be varied among ECU and receiver combinations, and we will take $K = 1$ for analysis. We can relate the parameters that we will extract to ω_n and ζ using the following relationships, shown in eq. (10), (11).

$$\zeta = \frac{|\ln \frac{\%OS}{100}|}{\sqrt{\pi^2 + (\ln \frac{\%OS}{100})^2}} \tag{10}$$

$$\omega_n = \frac{\pi}{T_p\sqrt{1-\zeta^2}} \tag{11}$$

From these equations, we see that extracting T_p and $\%OS$ are theoretically enough to perfectly represent an ideal second-order system's transient response, however, we include settling time and steady-state value calculations to enhance the model's predictions since the pulse digitization can cause small errors in parameter calculation. In addition, the ECUs may not behave exactly as ideal second-order systems, so these additional parameters could provide the necessary information if the response does not exactly follow the second-order behavior.

The algorithm for calculating these parameters is non-trivial. We start by analyzing an individual pulse k from $E_{(i)}$, $y_{(i,k)}(n)$, with length T. First, we find the peak value and its corresponding sample index, as shown in eq. (12) and eq. (13).

$$V_{peak} = max(y_{(i,k)}(n)) \tag{12}$$

$$n_{peak} = \underset{n}{\operatorname{argmax}}\, y_{(i,k)}(n) \tag{13}$$

Also, we already know where the rising and falling edges occur from eq. (7). Now, we calculate the peak time by finding the number of samples from the sample before the rising edge occurs to the sample corresponding to the peak value and multiplying by the sampling period, as shown in eq. (14).

$$T_p = (n_{peak} - n_{rise})T_s \tag{14}$$

We can calculate the steady-state value, assuming that the pulse does eventually settle, by selecting some number of samples Q of $y_{(i,k)}(n)$ to average just before the falling edge occurs, as shown in eq. (15). For our analysis, we use $Q = 10$.

$$V_{ss} = \frac{1}{Q}\sum_{n=n_{fall}-Q}^{n_{fall}-1} x(n) \tag{15}$$

Now, we can compute percentage overshoot, represented as a decimal, as shown in eq. (16).

$$\%OS = \frac{V_{peak} - V_{ss}}{V_{ss}} \tag{16}$$

Lastly, we can find settling time using a bit more complex of an algorithm. We define a kernel window of length J. We begin by aligning the beginning of the kernel window with the peak index, n_{peak}, of $y_{(i,k)}(n)$, and we check whether all values of $y_{(i,k)}(n)$ within the kernel window have magnitudes less than a threshold, ρ. If not, we slide the kernel one sample ahead until either this condition is met, which means we have found the settling index, or until the end of the kernel reaches the falling edge of the pulse, indicating the pulse does not settle. To find the settling time of a single pulse, first, we select a parameter, ρ, which is the percentage tolerance that a pulse voltage must remain from its steady-state value to be considered settled. Now, a tolerance value β can be calculated as $\beta = \rho V_{ss}$. A pulse has settled at index $n_{settled}$ when the following condition shown in eq. (17) has been satisfied.

$$V_{ss} - \beta \leq y_{(i,k)}(n) \leq V_{ss} + \beta, \quad n \geq n_{settled} \tag{17}$$

We can re-write eq. (17) as shown in eq. (18):

$$|y_{(i,k)}(n) - V_{ss}| \leq \beta, \quad n \geq n_{settled} \tag{18}$$

The following shows a more mathematical formulation of this algorithm. Allow n_0 to denote the first sample index in which the kernel window overlaps. We also will define $y^{(0)}_{(i,k)}(n)$, which is the same signal as $y_{(i,k)}(n)$ but with the D.C. steady-state offset removed.

$n_0 \leftarrow n_{peak}$
$y^{(0)}_{(i,k)}(n) \leftarrow y_{(i,k)}(n) - V_{ss}$
while $|y^{(0)}_{(i,k)}(n)| > \beta$ for all $n \in \{n_0, ..., n_0 + J - 1\}$ **do**
  $n_0 \leftarrow n_0 + 1$
  if $n_0 + J - 1 == n_{fall}$ **then**
    break
  end if
end while
$n_{settle} \leftarrow n_0$

Now that the settling index has been found, settling time is calculated by eq. (19).

$$T_s = (n_{settle} - n_{peak})T_s \tag{19}$$

The calculated parameters now comprise a lower-dimensional representation of a single-step response and can be used for pulse classification.

E. Training Neural Network:

After finding the feature-set, any supervised machine learning method including support vector machines, artificial neural networks, deep learning, etc. can be used to achieve this goal. In this research, we used an artificial neural network, implementing the batch gradient descent optimization algorithm to train using feature vectors from the database. An artificial neural network (ANN)-based model ($Fig.$ 15) is used to identify the source transmitter. For this the ANN gets $X_{(r)} = \{\mathbf{x}_{(1)}, \mathbf{x}_{(2)}, \ldots, \mathbf{x}_{(R)}\}$ as input-set and the output of neural network is the channel through which the signal

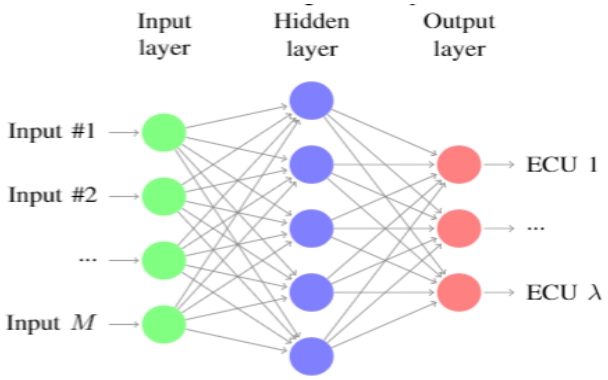

Fig. 15: *Artificial Neural Network Architecture*

propagated, which is further used to identify transmitter (ECU) for message authentication.

IV. Experimental Setup, Dataset and Results

A. Experimental Setup:

The proposed approach evaluates channel variability, to identify channel transmitter (ECU) for message authentication. Eight channels were used in this study and data was recorded through the $CANH$ pin. The technical specifications of channels are given in Table-I. The hardware is comprised of eight Arduino UNO-R2 micro-controller kits; eight CAN-Bus shield boards with $MCP\,2515$ CAN-bus controllers, $MPC\,2551$ CAN transceivers; and a $DSO1012A$ oscilloscope using a 2GSa/s sampling rate and $100\,MHz$ bandwidth to record the measured voltage samples. $Matlab\,R\,2018a$ software was used for the analysis of the recorded samples. A computer simulation was written that continuously transmitted the messages from different ECUs and pins. Afterward, these messages were then used as the dataset for model training and evaluation.

TABLE I: Technical specifications of channels

Label	Length(m)	Conductor	Insulation	Model
$Class$–1	2	Copper	XLPO	SAE J1939-15
$Class$–2	6	Copper	XLPO	SAE J1939-15
$Class$–3	10	Copper	XLPO	SAE J1939-15
$Class$–4	2	Copper	XLPO	SAE J1939-19
$Class$–5	6	Copper	XLPO	SAE J1939-19
$Class$–6	2	Copper	XLPO	SAE J1128
$Class$–7	6	Copper	XLPO	SAE J1128
$Class$–8	10	Copper	XLPO	SAE J1128

B. Dataset Description:

A dataset consisting of CAN packets at the output of 8 channels is recorded using an oscilloscope. There are 8 CAN channels, and 991 records in total comprising the dataset. For model evaluation, 70% of the dataset is randomly partitioned into a training set and 30% into testing set respectively. All data used is collected under the same controlled conditions i.e. under the same temperature and using identical message patterns to observe the unique behaviors of the sampled signals.

TABLE II: Confusion Matrix for Transmitter Classifier

		Target Class								
-	-	$Class$–1	$Class$–2	$Class$–3	$Class$–4	$Class$–5	$Class$–6	$Class$–7	$Class$–8	Acc. %
Predicted Class	$Class$–1	115	0	0	8	0	0	1	0	92.7
	$Class$–2	0	116	0	0	2	0	0	0	98.3
	$Class$–3	0	0	118	0	0	0	0	0	100
	$Class$–4	9	0	0	114	0	0	0	0	92.7
	$Class$–5	0	3	0	0	125	0	0	0	97.7
	$Class$–6	1	0	0	0	0	122	2	0	97.6
	$Class$–7	0	0	0	0	0	0	129	0	100
	$Class$–8	0	0	0	0	0	0	0	126	100
	Acc. %	92	97.5	100	93.4	98.4	99.2	98.5	100	97.4

C. Performance Evaluation Measures*:*

We used precision, recall, F_1–$Score$, accuracy and error rate as performance evaluation measures. The effectiveness of the method proposed is determined by the rate at which transmitters were correctly identified in the response to messages sniffed by $E_{(FP)}$. Let TP represent the number of true positive predictions, FP represents the number of false-positive predictions, TN represents the number of true negative predictions, and FN represents the number of false-negative predictions. We computed precision, recall, F_1–$Score$, accuracy and error rate as follows:

$$Precision = \left(\frac{TP}{TP + FP}\right) \tag{20}$$

$$Recall = \left(\frac{TP}{TP + FN}\right) \tag{21}$$

$$F_1\text{–}score = 2 \times \left(\frac{Precision \times Recall}{Precision + Recall}\right) \tag{22}$$

$$Accuracy = \left(\frac{TP + TN}{TP + TN + FP + FN}\right) \tag{23}$$

$$Error\text{–}rate = 1 - Accuracy \tag{24}$$

D. Experimental Results and Analysis:

A series of experiments are performed for performance evaluation. For the performance evaluation presented, a neural network classifier is trained on the feature vectors of all different channels. It is important to highlight here that the selection of neural networks for the classifier is just a matter of choice and not a limitation of the proposed method. Classification accuracy is used as a performance measure here. Table-II shows that the method proposed achieves a correct detection rate of 97.4%. $Fig.$ 16 gives a visual representation of various useful metrics in the performance measurement of the classifier. The ideal classifier will have a precision, recall, accuracy, and F_1–$Score$ of 100%, and an error of 0%. Table-III shows the numerical performance metrics per ECU-channel combination. We see relatively good performance when looking at the accuracy and error scores, which only considers whether each prediction was correct or not. Thus, the accuracy and error consider TN and TP both as correct predictions, and FN and FP predictions as incorrect. These are useful metrics to get an idea for how the model performs, but they do not give much insight into where the failures in the model exist. Precision and recall scores are useful for this purpose. The F_1–$Score$ combines the recall and precision into one score, which gives another comprehensive summary of model performance.

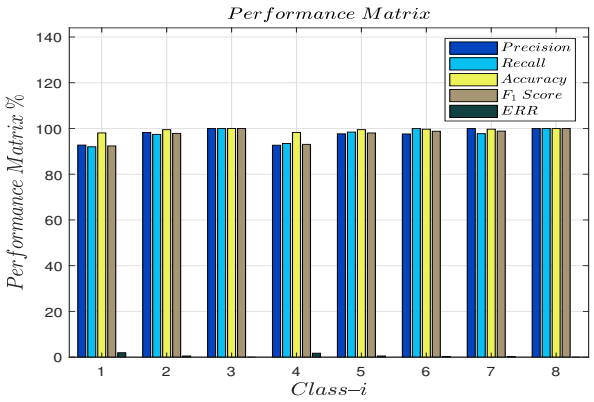

Fig. 16: $Bar\ graph\ of\ PM\ for\ Channel\ Classifier$

The receiver operating characteristic curve, shown in $Fig.$ 17, depicts the true-positive rate (TPR) and false-positive rate (FPR) plotted against each other. In the ideal case in which we have a perfect classifier, the TPR will be 100% and FPR will be 0%. This means that the area under the ROC curve is 1 in the ideal case. With an imperfect classifier, this curve will begin to bend such that the area under the curve is reduced. Comparing the area under the ROC curve is one way to determine which classes were predicted best and worst

TABLE III: $Performance\ Matrix\ of\ ECU\ Classifier$

-	$Precision$	$Recall$	$Accuracy$	F_1–$Score$	ERR
$Class$–1	92.7%	92.0%	98.1%	92.4%	1.9%
$Class$–2	98.3%	97.4%	99.5%	97.9%	0.5%
$Class$–3	100%	100%	100%	100%	0.0%
$Class$–4	92.7%	93.4%	98.3%	93.1%	1.7%
$Class$–5	97.7%	98.4%	99.5%	98.0%	0.5%
$Class$–6	97.6%	100%	99.7%	98.8%	0.3%
$Class$–7	100%	97.7%	99.7%	98.8%	0.3%
$Class$–8	100%	100%	100%	100%	0%

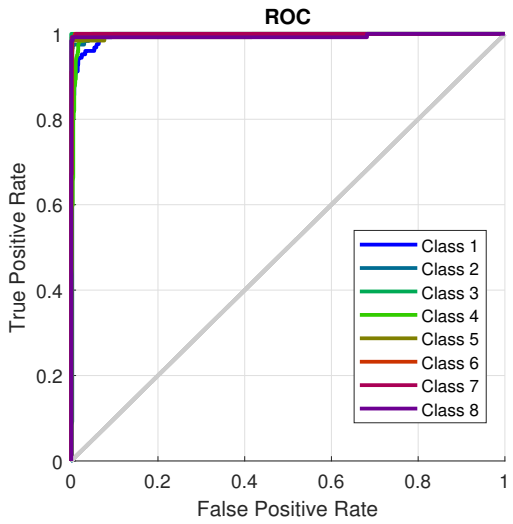

Fig. 17: *ROC of Class–i*

by the model. Using *Fig.* 17, we see that *Class*–1 is the least similar to the ideal case. Note that this bend in the curve for *Class*–1 shows that initially if we have no false positives, we will not have a high true positive rate. To improve our ability to predict *Class*–1 signals as being from *Class*–1, we also must sacrifice our FPR, so we will increase the chance of a signal from any other class being predicted as *Class*–1 as well. The other classes show that they can achieve very high TPR without much sacrifice in FPR.

E. Robustness against Spoofing attacks:

Our method is robust against ECU impersonation attacks that are launched through the physical access of the car e.g. the adversary attacks the CAN through physical access by deploying a new ECU. Since the distortion in the signal is dependent on DAC and material imperfections, the feature vector estimated will be different as well, the message will not be authenticated and attack will be identified. Moreover, our method is also effective for attacks launched through wireless interface e.g. attack launched on the infotainment system to access CAN as done by Miller et. al. [1]. Given a vehicle network as shown in *Fig.* 18, suppose that an adversary tries to penetrate the vehicle network through the wireless interface of the infotainment system with the help of a CAN message, which is for the braking system. In CAN message, the information about the sender is missing, as the messages are functionality-based. However, the fingerprinting ECU $E_{(FP)}$ will correctly recognize the ECU, it will not authenticate this message as it is coming from the wrong sender, and it will send a warning signal to braking unit ECU. Now, the ECU braking unit knows not to apply the brakes since the CAN message for applying brakes is not supposed to come from the infotainment system.

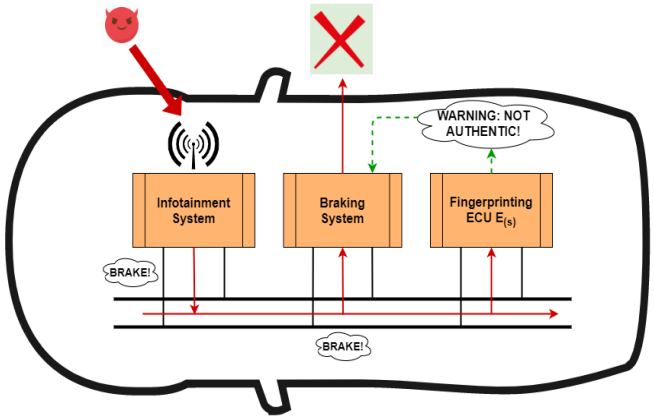

Fig. 18: *Effectiveness of our method in case of attack for in–vehicle Communication*

V. Conclusion

The main contribution of this paper is that each channel in CAN injects a unique signature in the physical signal, it has been demonstrated that the channel-specific step response in the received signal is unique; hence, it can be utilized to associate CAN packets with their source transmitter. The proposed system uses the transient response parameters of the channel to capture uniqueness in the received signal, which is used as inputs to a multi-layer perceptron neural network classifier. The classification method's performance is evaluated on eight different CAN channels. The experimental results indicate that each channels' set of feature vectors is significantly different from all other channels. The proposed method achieves channel detection with an accuracy of 97.4%. In the future, we plan to add more channels and ECUs to the network. We also plan to test our system in different environments by varying factors such as temperature, humidity, etc. We also plan to test the performance of our system when subjected to different electromagnetic interference. Channel response can be used to localize the source i.e. transmitter. Hence, in case of attack, the adversary can be localized and the adversary ECU will be set to bus-off state from the network. In this way, the attacker will not be able to launch spoofing attacks once the source of the attack is localized.

REFERENCES

[1] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," *Black Hat USA*, p. 91, 2015.

[2] S. Liu, "Cybersecurity Market Revenues Worldwide 2017-2023," https://www.statista.com, 2019.

[3] "Automotive Revolution - Perspective Toward 2030," https://www.mckinsey.com, 2016.

[4] B. Gierlichs and A. Y. Poschmann, *Cryptographic Hardware and Embedded Systems–CHES 2016*. Springer, 2016.

[5] A. Hazem and H. Fahmy, "Lcap-a lightweight can authentication protocol for securing in-vehicle networks," in *10th escar Embedded Security in Cars Conference, Berlin, Germany*, vol. 6, 2012.

[6] T. P. Doan and S. Ganesan, "Can crypto fpga chip to secure data transmitted through can fd bus using aes-128 and sha-1 algorithms with a symmetric key," SAE Technical Paper, Tech. Rep., 2017.

[7] H. Ueda, R. Kurachi, H. Takada, T. Mizutani, M. Inoue, and S. Horihata, "Security authentication system for in-vehicle network," *SEI Technical Review*, vol. 81, pp. 5–9, 2015.

[8] T. Sugashima, D. K. Oka, and C. Vuillaume, "Approaches for secure and efficient in-vehicle key management," *SAE International Journal of Passenger Cars-Electronic and Electrical Systems*, vol. 9, no. 2016-01-0070, pp. 100–106, 2016.

[9] A. Hafeez, H. Malik, O. Avatefipour, P. R. Rongali, and S. Zehra, "Comparative study of can-bus and flexray protocols for in-vehicle communication," SAE Technical Paper, Tech. Rep., 2017.

[10] M. Wolf, A. Weimerskirch, and C. Paar, "Security in automotive bus systems," in *Workshop on Embedded Security in Cars*, 2004.

[11] P.-S. Murvay and B. Groza, "Source identification using signal characteristics in controller area networks," *IEEE Signal Processing Letters*, vol. 21, no. 4, pp. 395–399, 2014.

[12] R. M. Gerdes, M. Mina, S. F. Russell, and T. E. Daniels, "Physical-layer identification of wired ethernet devices," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 4, pp. 1339–1353, 2012.

[13] O. Avatefipour, "Physical-fingerprinting of electronic control unit (ecu) based on machine learning algorithm for in-vehicle network communication protocol "can-bus"," 2017.

[14] Q. Wang and S. Sawhney, "Vecure: A practical security framework to protect the can bus of vehicles," in *Internet of Things (IOT), 2014 International Conference on the*. IEEE, 2014, pp. 13–18.

[15] K.-T. Cho and K. G. Shin, "Viden: Attacker identification on in-vehicle networks," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 1109–1123.

[16] ——, "Fingerprinting electronic control units for vehicle intrusion detection." in *USENIX Security Symposium*, 2016, pp. 911–927.

[17] W. Choi, K. Joo, H. J. Jo, M. C. Park, and D. H. Lee, "Voltageids: Low-level communication characteristics for automotive intrusion detection system," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 8, pp. 2114–2129, 2018.

[18] B. Groza and P.-S. Murvay, "Efficient intrusion detection with bloom filtering in controller area networks (can)," *IEEE Transactions on Information Forensics and Security*, 2018.

[19] S. U. Sagong, X. Ying, A. Clark, L. Bushnell, and R. Poovendran, "Cloaking the clock: emulating clock skew in controller area networks," in *Proceedings of the 9th ACM/IEEE International Conference on Cyber-Physical Systems*. IEEE Press, 2018, pp. 32–42.

[20] W. Wu, R. Li, G. Xie, J. An, Y. Bai, J. Zhou, and K. Li, "A survey of intrusion detection for in-vehicle networks," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[21] M. L. Han, J. Lee, A. R. Kang, S. Kang, J. K. Park, and H. K. Kim, "A statistical-based anomaly detection method for connected cars in internet of things environment," in *International Conference on Internet of Vehicles*. Springer, 2015, pp. 89–97.

[22] H. M. Song, H. R. Kim, and H. K. Kim, "Intrusion detection system based on the analysis of time intervals of can messages for in-vehicle network," in *2016 international conference on information networking (ICOIN)*. IEEE, 2016, pp. 63–68.

[23] A. Taylor, N. Japkowicz, and S. Leblanc, "Frequency-based anomaly detection for the automotive can bus," in *2015 World Congress on Industrial Control Systems Security (WCICSS)*. IEEE, 2015, pp. 45–49.

[24] H. Lee, S. H. Jeong, and H. K. Kim, "Otids: A novel intrusion detection system for in-vehicle network by using remote frame," in *2017 15th Annual Conference on Privacy, Security and Trust (PST)*. IEEE, 2017, pp. 57–5709.

[25] M. Marchetti, D. Stabili, A. Guido, and M. Colajanni, "Evaluation of anomaly detection for in-vehicle networks through information-theoretic algorithms," in *2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI)*. IEEE, 2016, pp. 1–6.

[26] W. Wu, Y. Huang, R. Kurachi, G. Zeng, G. Xie, R. Li, and K. Li, "Sliding window optimized information entropy analysis method for intrusion detection on in-vehicle networks," *IEEE Access*, vol. 6, pp. 45 233–45 245, 2018.

[27] D. Stabili, M. Marchetti, and M. Colajanni, "Detecting attacks to internal vehicle networks through hamming distance," in *2017 AEIT International Annual Conference*. IEEE, 2017, pp. 1–6.

[28] A. Taylor, S. Leblanc, and N. Japkowicz, "Anomaly detection in automobile control network data with long short-term memory networks," in *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 2016, pp. 130–139.

[29] M.-J. Kang and J.-W. Kang, "Intrusion detection system using deep neural network for in-vehicle network security," *PloS one*, vol. 11, no. 6, p. e0155781, 2016.

[30] M. Markovitz and A. Wool, "Field classification, modeling and anomaly detection in unknown can bus networks," *Vehicular Communications*, vol. 9, pp. 43–52, 2017.

[31] N. Jain and S. Sharma, "The role of decision tree technique for automating intrusion detection system," *International Journal of Computational Engineering Research (ijceronline. com)*, vol. 2, no. 4, 2012.

[32] R. Rieke, M. Seidemann, E. K. Talla, D. Zelle, and B. Seeger, "Behavior analysis for safety and security in automotive systems," in *2017 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*. IEEE, 2017, pp. 381–385.

[33] S. N. Narayanan, S. Mittal, and A. Joshi, "Using data analytics to detect anomalous states in vehicles," *arXiv preprint arXiv:1512.08048*, 2015.

[34] M. Marchetti and D. Stabili, "Anomaly detection of can bus messages through analysis of id sequences," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 1577–1583.

[35] A. Hafeez, M. Tayyab, C. Zolo, and S. Awad, "Finger printing of engine control units by using frequency response for secure in-vehicle communication," in *2018 14th International Computer Engineering Conference (ICENCO)*. IEEE, 2018, pp. 79–83.

[36] R. M. Gerdes, M. Mina, S. F. Russell, and T. E. Daniels, "Physical-layer identification of wired ethernet devices," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 4, pp. 1339–1353, 2012.

[37] W. Choi, H. J. Jo, S. Woo, J. Y. Chun, J. Park, and D. H. Lee, "Identifying ecus using inimitable characteristics of signals in controller area networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 6, pp. 4757–4770, 2018.

[38] M. Kneib and C. Huth, "Scission: Signal characteristic-based sender identification and intrusion detection in automotive networks," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2018, pp. 787–800.

[39] O. Avatefipour, A. Hafeez, M. Tayyab, and H. Malik, "Linking received packet to the transmitter through physical-fingerprinting of controller area network," in *Information Forensics and Security (WIFS), 2017 IEEE Workshop on*. IEEE, 2017, pp. 1–6.

[40] J. Hall, M. Barbeau, and E. Kranakis, "Radio frequency fingerprinting for intrusion detection in wireless networks," *IEEE Transactions on Defendable and Secure Computing*, vol. 12, pp. 1–35, 2005.

[41] M. Tayyab, A. Hafeez, and H. Malik, "Spoofing attack on clock based intrusion detection system in controller area networks," in *2018 Ground Vehicle Systems Engineering and Technology Symposium, Proceedings, Novi, Michigan*. GVSETS, 2018.

[42] A. Hafeez, H. Malik, and K. Mahmood, "Performance of blind microphone recognition algorithms in the presence of anti-forensic attacks," in *Audio Engineering Society Conference: 2017 AES International Conference on Audio Forensics*. Audio Engineering Society, 2017.

[43] A. Hafeez, K. M. Malik, and H. Malik, "Exploiting frequency response for the identification of microphone using artificial neural networks," in *Audio Engineering Society Conference: 2019 AES INTERNATIONAL CONFERENCE ON AUDIO FORENSICS*. Audio Engineering Society, 2019.

[44] A. Van Herrewege, D. Singelee, and I. Verbauwhede, "Canauth-a simple, backward compatible broadcast authentication protocol for can bus," in *ECRYPT Workshop on Lightweight Cryptography*, vol. 2011, 2011.

[45] F. T. Ulaby, E. Michielssen, and U. Ravaioli, *Fundamentals of Applied Electromagnetics 6e*. Prentice Hall, 2001.